

Interpolación en Matlab

Ejemplo:

```
N = 10
```

```
x = 1:N
```

```
y = -3*x.^2 + 1.7.*x - 17;
```

```
figure; plot(x,y,'+')
```

```
p = polyfit(x,y,N-1)
```

```
v = polyval(p,1)
```

En la realidad los datos están “sucios”

Ejemplo:

$$y = -3*x.^2 + 1.7.*x - 17 + \text{rand}(1, 10)*20;$$

Recursividad

Recursividad

- Definición de factorial

- $n! = n * (n-1)! \quad n > 1, \text{ con } 1! = 1$

```
function v = fact(n)
    if n <= 1
        v = 1;
    else
        v = n * fact(n-1);
    end
end
```

Los conejos y la serie de Fibonacci

La serie de Fibonacci indica cuantas parejas adultas hay en un mes dado



Si en un mes se tienen **a** parejas jóvenes y **b** parejas adultas, al siguiente mes se tendrán **a + b** parejas adultas y **b** parejas jóvenes. Por lo tanto, el número de conejos adultos en un mes **n**, es el número de conejos adultos en el mes **n-1** más el número de conejos jóvenes en el mes **n-1**.

Como el número de conejos jóvenes en el mes **n-1** es el número de conejos adultos en el mes **n-2**, entonces podemos concluir que

$$F(0) = 0$$

$$F(1) = 1$$

$$F(n) = F(n - 1) + F(n - 2)$$

Version recursiva

$$F(0) = 0$$

$$F(1) = 1$$

¿cuantas veces se hace llamar a la funcion?

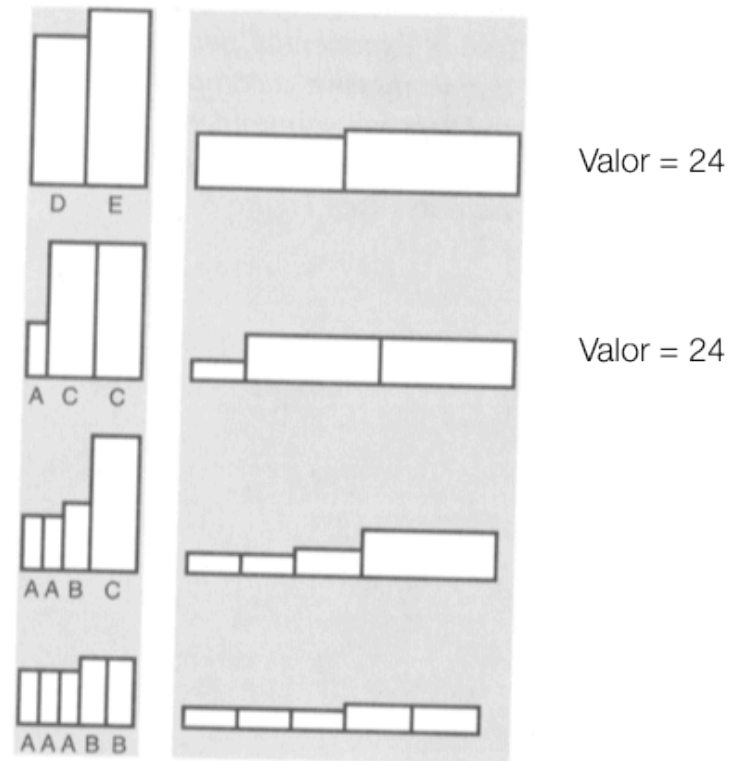
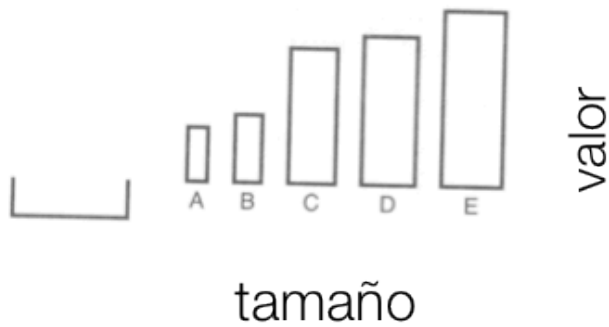
Version iterativa

Usando un vector

Usando 2 variables

El Problema de la bolsa del ladron

	0	1	2	3	4
item	A	B	C	D	E
size	3	4	7	8	9
val	4	5	10	11	13



Para una bolsa de tamaño 17

Variables que necesitamos

```
function main_knapsack
```

```
names(1) = 'A';  names(2) = 'B';  names(3) = 'C';  
names(4) = 'D';  names(5) = 'E';
```

```
sizes(1) = 3;  sizes(2) = 4;  sizes(3) = 7;  
sizes(4) = 8;  sizes(5) = 9;
```

```
values(1) = 4;  values(2) = 5;  values(3) = 10;  
values(4) = 11;  values(5) = 13;
```

Variables que necesitamos

```
function main_knapsack
```

```
names(1) = 'A'; names(2) = 'B'; names(3) = 'C';  
names(4) = 'D'; names(5) = 'E';
```

```
sizes(1) = 3; sizes(2) = 4; sizes(3) = 7;  
sizes(4) = 8; sizes(5) = 9;
```

```
values(1) = 4; values(2) = 5; values(3) = 10;  
values(4) = 11; values(5) = 13;
```

Variables que necesitamos (leerlos de archivo)

Bajar los archivos de datos de:

http://www.cimat.mx/~alram/info_apli2/valores_objetos.txt

http://www.cimat.mx/~alram/info_apli2/datos_objetos.txt

```
% Para leer los puros valores
```

```
objs = load('c:\documentos\valores_objetos.txt')
```

```
sizes = objs(:,1)
```

```
values = objs(:,2)
```

```
% Para leer todos los datos
```

```
[names sizes values] = textread('c:\datos_objetos.txt', '%s %f %f')
```

Funcion recursiva

```
function maxVal = myKnapsack(cap,nObjects,sizes,values)

    maxVal = 0; % max for this capacity
    for i=1:nObjects % we select ith element

        spaceLeft = cap - sizes(i);

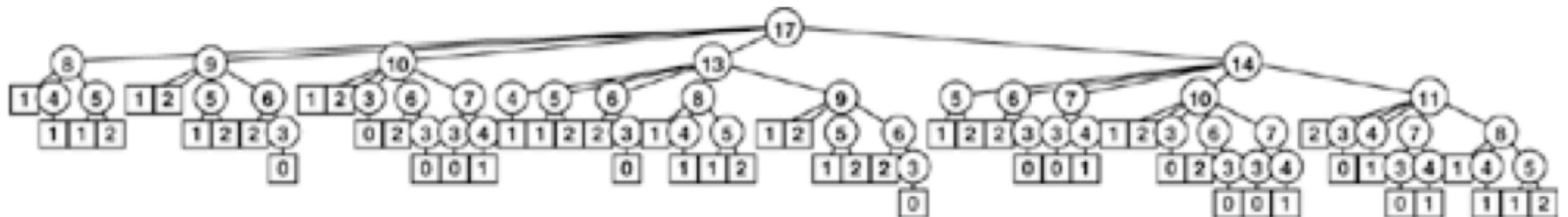
        if spaceLeft >= 0
            maxValTemp = values(i) + myKnapsack(spaceLeft,nObjects,sizes,values);
            if maxValTemp > maxVal
                maxVal = maxValTemp;
            end
        end

    end

end

end
```

Arbol de llamadas recursivas para tamaño 17



	0	1	2	3	4
item	A	B	C	D	E
size	3	4	7	8	9
val	4	5	10	11	13

